

## 3 Protótipos

Nesta seção apresentamos protótipos que desenvolvemos com o objetivo de levantar os requisitos necessários para um sistema para apresentações multimídia distribuídas. Os protótipos auxiliaram na construção da infra-estrutura de execução apresentada no Capítulo 4 e também na definição dos requisitos da representação do documento multimídia distribuído apresentado no Capítulo 5.

Neste capítulo discutiremos protótipos que integram aplicações como o Microsoft PowerPoint® e o Windows Media Player®, dois programas comerciais de código fechado que oferecem automação através de COM [8] e, portanto, nos permitem apenas o controle externo de execução.

Uma adaptação da interface COM dessas aplicações nos permitiu distribuí-las através de CORBA [61]. Utilizamos para essa adaptação a linguagem de colagem LuaOrb [9, 17], que facilita a interoperabilidade entre diferentes middlewares, convertendo tipos de dados e oferecendo rápida prototipação.

Além dessas aplicações comerciais, integramos o Navegador 3D, uma ferramenta de código aberto construída para permitir ao usuário a navegação em modelos tridimensionais [50]. Utilizamos também um sensor chamado Ubisensor [1], capaz de captar a posição espacial de *sensores* através de tecnologia de banda de alta frequência.

### 3.1 Primeiro Protótipo

O Microsoft PowerPoint®, como componente de uma apresentação, se mostrou bastante atraente pela facilidade de composição de conteúdo. Esta característica, aliada à facilidade de uso pelos usuários, a torna uma ferramenta interessante tanto para o trabalho colaborativo quanto para a composição de apresentações multimídia.



Figura 3.1: Uma apresentação PowerPoint® sendo acompanhada em um emulador de aparelho celular

Nosso objetivo principal com este protótipo era experimentar a flexibilidade de controle dos componentes da apresentação através da infraestrutura de execução. Para tanto, testamos controladores de grupo variados, como aplicações comuns, páginas Web, reconhecedores de voz, aparelhos de celular e a própria interface com o usuário que o PowerPoint® oferece.

A interface de automação do PowerPoint® nos permitiu obter bons resultados de integração com a infra-estrutura *ActivePresentation*. Funcionalidades como início e término de aplicações, transferência de arquivos e controle em grupo nos auxiliaram na construção de um sistema de trabalho colaborativo no qual é feita uma execução sincronizada entre máquinas, permitindo operação em conjunto através de múltiplos controladores.

Em um ambiente de trabalho com debates e interação, o controle distribuído de uma apresentação PowerPoint® pode obedecer a seguinte semântica: todas as operações de troca de slides são enviadas para o grupo e o slide final será o resultante da união de todas as ações.

Através da ferramenta Brew [21] de desenvolvimento de aplicativos para celulares CDMA, construímos um visualizador e controlador diferente para grupos de PowerPoints®. Uma imagem do protótipo é vista na Figura 3.1, na qual um emulador Brew mostra os controles e a imagem de uma apresentação PowerPoint® em execução.

Por ser uma ferramenta para dispositivos móveis, Brew oferece uma infra-estrutura de software reduzida, permitindo a comunicação entre dis-

positivos apenas através de sockets. Desta forma, para integrar o aparelho de celular à infra-estrutura *ActivePresentation* utilizamos um serviço intermediário que, através de um simples protocolo de comunicação com o aparelho de celular, integra os dois sistemas.

### 3.2

#### Segundo Protótipo

Após a construção do primeiro protótipo com o Microsoft PowerPoint®, integramos o Windows Media Player® utilizando a mesma infra-estrutura de execução. O objetivo de integrar o Windows Media Player® foi utilizá-lo como componente de reprodução de áudio e vídeo, peça fundamental para a composição de apresentações multimídia. Como ocorreu com o Microsoft PowerPoint®, utilizamos a interface de automação COM oferecida por este componente para controlar e observar eventos ocorridos no decorrer da sua execução.

Nosso objetivo com este protótipo foi, em primeiro lugar, averiguar a generalidade da infra-estrutura. Através dos bons resultados obtidos, principalmente pela agilidade com que o Windows Media Player® foi inserido na infra-estrutura, concluímos que obtivemos êxito neste propósito. Nosso segundo objetivo era interligar os dois protótipos de forma a construir uma apresentação multimídia distribuída.

Para interligar as aplicações, construímos uma terceira aplicação para coordenar a execução do PowerPoint® em sincronia com a reprodução do vídeo. Utilizamos o formato de vídeo ASF [40], que permite inserir marcas de tempo nos arquivos de vídeo as quais são reconhecidas pelo Windows Media Player®. O Windows Media Player® informava através de sua interface de automação quando essas marcas eram atingidas durante a reprodução de um vídeo. Através da notificação da ocorrência desses eventos pela infra-estrutura *ActivePresentation*, a aplicação controladora mudava o *slide* do PowerPoint®.

### 3.3

#### Terceiro Protótipo

Ambientes tridimensionais presentes em jogos, simulações e projetos de engenharia já fazem parte do cotidiano de trabalho de muitas empresas. Em nosso terceiro protótipo, investigamos o uso de aplicações de visualização 3D em apresentações multimídia distribuídas. Para isso, utilizamos o

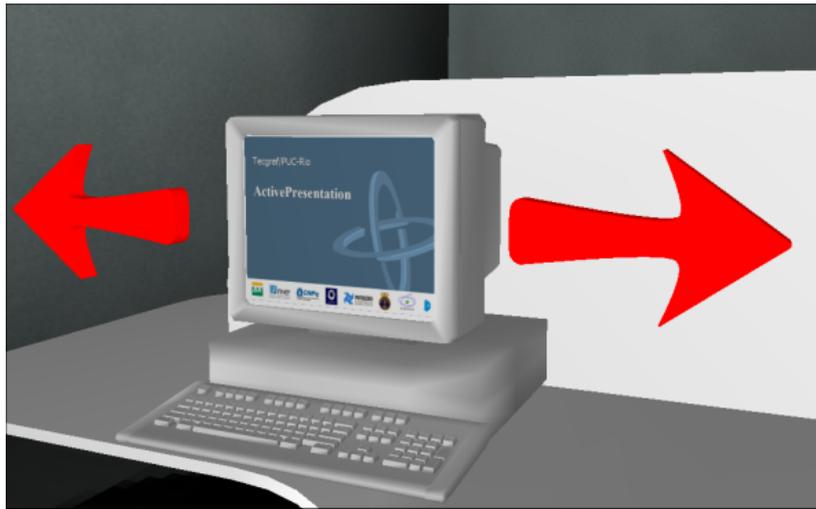


Figura 3.2: Ambiente Virtual de visualização e controle de uma apresentação PowerPoint® real

*Navegador 3D* desenvolvido pelo Tecgraf/PUC-Rio, que é uma aplicação de visualização tridimensional que oferece facilidades como navegação, carga de modelos e interação sobre objetos 3D [50].

Em nosso protótipo, inserimos o *Navegador 3D* como uma aplicação dentro de nossa infra-estrutura. A Figura 3.2 mostra um modelo tridimensional de um computador que executa um PowerPoint® fictício. A imagem de um *slide*, vista nessa Figura, é retirada de uma apresentação que ocorre em um computador real.

A apresentação PowerPoint® real pode ser manipulada através das setas virtuais, que podem ser selecionadas dentro do ambiente virtual. Além disso, quando ocorre uma troca de slide na apresentação, a imagem exibida no ambiente virtual é atualizada.

Futuramente, essa idéia pode ser estendida para controlar remotamente sistemas complexos através de uma representação virtual do sistema, oferecendo um alto grau de semelhança com a interface real.

### 3.4 Quarto Protótipo

Em nosso quarto protótipo experimentamos o uso de um sistema comum em ambientes de computação ubíqua: um sistema de localização. Utilizamos um sensor chamado Ubisensor [1], que capta valores tridimensionais da posição de receptores através de sinais de rádio de alta frequência.

A nossa idéia é interligar informações do mundo real com uma representação virtual. Esta ligação é possível através do Ubisensor e do naveg-

ador 3D que respectivamente obtém o posicionamento real de um receptor e representa em um ambiente de visualização tridimensional. Nosso objetivo principal era obter informações quanto à dificuldade de representação deste cenário em um documento multimídia.

Para adequar a visualização do modelo ao sistema de posicionamento, criamos um espaço virtual nas mesmas proporções do espaço físico. Além disso, colocamos o modelo em uma posição que refletia as mesmas coordenadas que eram enviadas pelo sistema de posicionamento. Por exemplo, se havia um canto da mesa na posição (2, 3, 4) fornecida pelo Ubisensor, então no modelo virtual, esta coordenada deveria ser a mesma.

Em nosso experimento, utilizamos uma aplicação para obter a lista dos receptores e constantemente verificar a posição deles. A posição consultada era refletida em uma chamada remota no navegador 3D para ajustar a posição dos sensores no modelo tridimensional.

Como resultado desse experimento, constatamos que é possível integrar as duas aplicações. Verificamos que no sistema de posicionamento as chamadas remotas para a obtenção da posição dos sensores exigiam o identificador do sensor. Na aplicação de visualização, a função de alteração da posição do sensor no modelo exigia o identificador do receptor e a sua nova posição. Portanto, percebemos que para a representação dessa ligação em um documento multimídia, precisamos da descrição do receptor através de seu código identificador e da representação de coordenadas tridimensionais.

Como cada aplicação irá oferecer uma API de controle totalmente diferente, concluímos que documentos multimídia não devem tentar fazer referência à API das aplicações, mas devem representar os estados das aplicações e as relações entre os estados. Em função disso, cada aplicação deve oferecer um adaptador de controle que recebe os estados de execução a serem observados. O software que irá interpretar o documento multimídia e garantir as ligações ou restrições de sincronia passará essas informações a cada aplicação. Essa abordagem é similar ao feita no trabalho de Rogério Ferreira Rodrigues e Luiz Fernando Gomes Soares na implementação do interpretador e executor de documentos NCL, o *formatador* HyperProp [12, 22].

### 3.5

#### Quinto Protótipo

A partir das conclusões do quarto protótipo (Seção 3.4), descrevemos uma apresentação através de uma versão experimental de um documento

multimídia. A apresentação era composta de um grupo de vídeos sincronizados entre si e *slides* apresentados por um grupo de PowerPoints®. O vídeo dirigia a apresentação e através da sua posição os slides eram mudados.

Além da descrição do documento, desenvolvemos um formator, isto é, um software capaz de interpretar e executar essa apresentação. O objetivo deste protótipo era verificar a dificuldade de representação, construir uma versão simplificada de um formatador e verificar a flexibilidade da infraestrutura de execução.

Um dos obstáculos desse protótipo foi a sincronização dos vídeos em diferentes computadores. Um software para a sincronização dos vídeos precisou ser especificamente desenvolvido para o Windows Media Player®. Devido ao fato dessa solução ser válida apenas para o Windows Media Player®, concluímos que não seria adequado pensarmos em uma solução de representação desse tipo de sincronia em um documento multimídia.

Para experimentar a criação de formatadores de apresentações multimídia distribuídas fizemos uso do sistema HyperProp [12, 22] desenvolvido pelo grupo Telemídia da PUC-Rio. Integramos ao formatador um adaptador capaz de controlar a aplicação PowerPoint®. Porém, percebemos que, para incluir as aplicações que gostaríamos de experimentar, o formatador iria exigir constantes alterações. Desta forma, optamos pela implementação de um formatador simples para experimentar nossas idéias. Neste momento também optamos pelo uso da linguagem Lua [41] para a representação do documento, o motivo principal sendo devido ao fato do sistema de execução já estar escrito nesta linguagem.

O desenvolvimento de um formator simplificado para a interpretação do documento multimídia permitiu investigar a eficácia da representação através de um sistema funcional. Uma das conclusões tiradas foi a necessidade de indicar os dispositivos que iriam executar cada mídia. Além disso, notamos que a verificação da posição de reprodução do vídeo exigia que o sistema fizesse um *pooling* constante desse valor. Desta forma, avaliamos que idealmente o sistema de interpretação deveria repassar para cada componente de mídia a lista dos elos a serem observados; assim, cada componente ficaria responsável por informar quando um elo era atingido. Novamente, esta abordagem é similar à tomada pelo sistema HyperProp [12, 22] anteriormente mencionado. Portanto, como nosso foco é a representação do documento e a infra-estrutura de execução, ajustamos a infra-estrutura para que os componentes de mídia pudessem enviar mensagens genéricas entre as aplicações.